



Object-Based Image Editing

Alan S. Cheney

William A. Barrett

Brigham Young University



Image Editing Work

- Pixel-based Methods
 - Adobe Photoshop, GIMP, SuperGoo

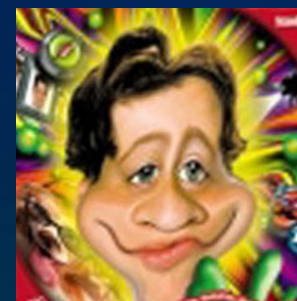




Image Editing Work

- Pixel-based Methods
 - Adobe Photoshop, GIMP, SuperGoo
- Image-based Methods
 - Wolberg '90-'00 Image Warping

Image Editing Work

- Pixel-based Methods
 - Adobe Photoshop, GIMP, SuperGoo
- Image-based Methods
 - Wolberg '90-'00 Image Warping
- Image-based Methods
 - Beier & Neely '92 Field Morphing
 - Wolberg '90-'00 Image Warping

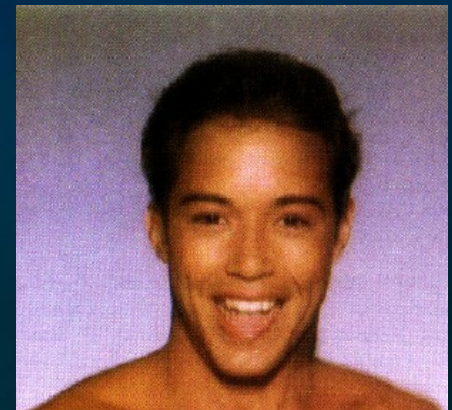


Image Editing Work

- Pixel-based Methods
 - Adobe Photoshop, GIMP, SuperGoo
- Image-based Methods
 - Wolberg '90-'00 Image Warping
 - Beier & Neely '92 Field Morphing
 - Lee, et. al '94 Thin-plate Splines

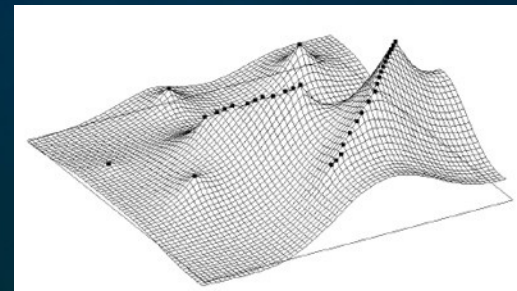




Image Editing Work

- Pixel-based Methods
 - Adobe Photoshop, GIMP, SuperGoo
- Image-based Methods
 - Wolberg '90-'00 Image Warping
 - Beier & Neely '92 Field Morphing
 - Lee, et. al '94 Thin-plate Splines
- Region-of-Interest Methods
 - Image-based methods could be applied to regions-of-interest

Image Editing Work

- Pixel-based Methods
 - Adobe Photoshop, GIMP, SuperGoo
- Image-based Methods
 - Wolberg '90-'00 Image Warping
 - Beier & Neely '92 Field Morphing
 - Lee, et. al '94 Thin-plate Spline
- Region-of-Interest Methods
 - Image-based methods could be applied to regions-of-interest
 - Elder & Goldberg '98, '01



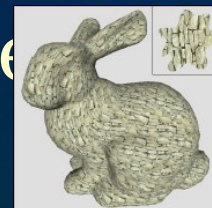
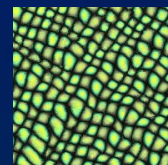
Image Editing Work

- Pixel-based Methods
 - Adobe Photoshop, GIMP, SuperGoo
- Image-based Methods
 - Wolberg '90-'00 Image Warping
 - Beier & Neely '92 Field Morphing
 - Lee, et. al '94 Thin-plate Spline
- Region-of-Interest Methods
 - Image-based methods could be applied to regions-of-interest
 - Elder & Goldberg '98, '01



Texture Synthesis Work

- Efros & Leung '99
Non-parametric Sampling
- Wei & Levoy 2000 Faster with TSVQ
- Harrison 2000 Resynthesizer plug-in
- Praun, et al. 2000 Lapped Textures
- Liang, et al. '01 Real-Time -
Patch-Based Sampling
- Efros & Freeman '01
Image Quilting





Where does Object-Based Image Editing fit in?

Pixel-based Methods




Image-based Methods



Where does Object-Based Image Editing fit in?

SuperGoo, clone tool, nudge tool, etc.



Object-Based Image Editing

Image Warping, Morphing, Spline warping



Video



What we're going to cover:



1. Object selection & representation
2. Object editing operations
3. Rendering
4. Background filling and texture painting
5. Applications

Object Selection

1. Segment image into *catchment basins* (TRAPs)
(*Tobogganed Regions of Accumulated Plateaus*)

using

as

algorithm.

TRAPs



Select
ed
Object

2. Manually “tag” TRAPs to select object



Object Representation

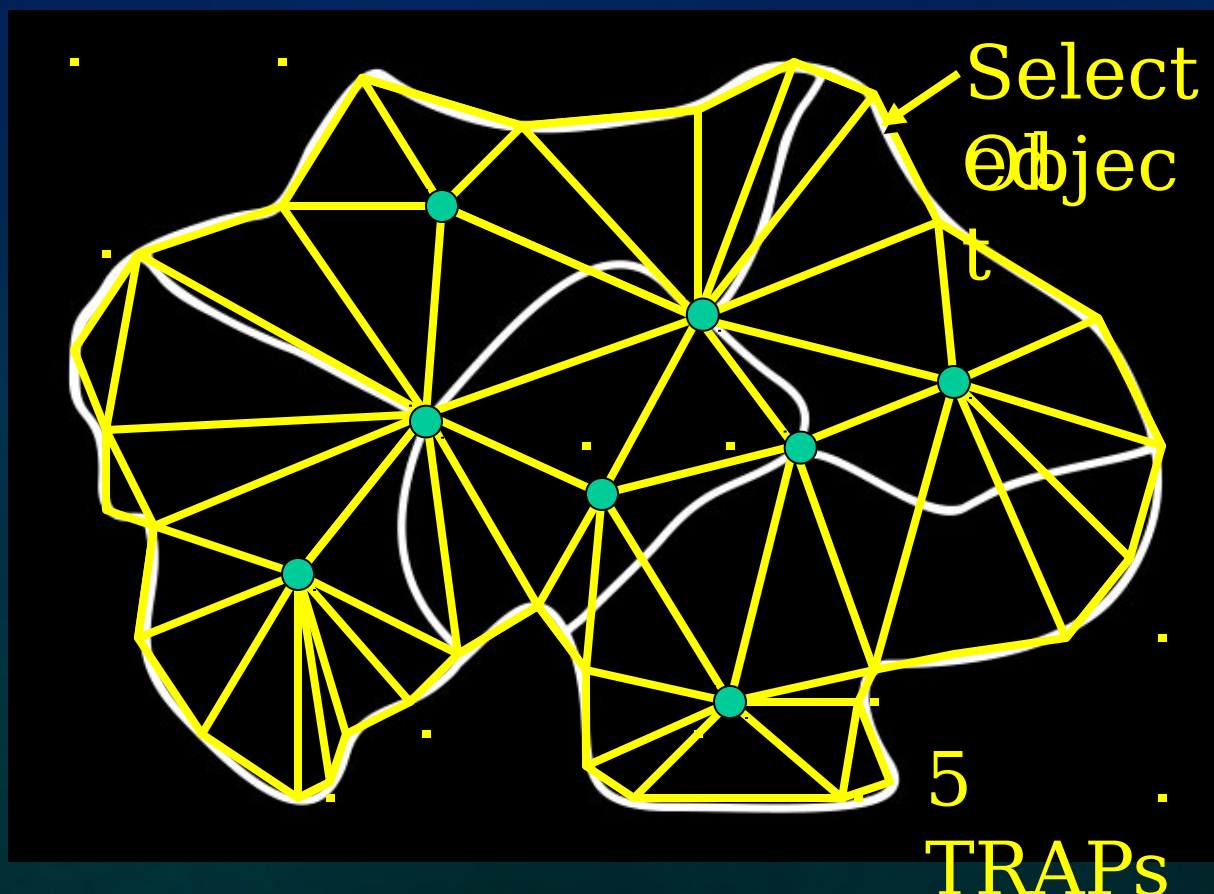
SAN ANTONIO
SIGGRAPH
2002

Fit Triangular Mesh to selected Object
- for efficient OpenGL Rendering



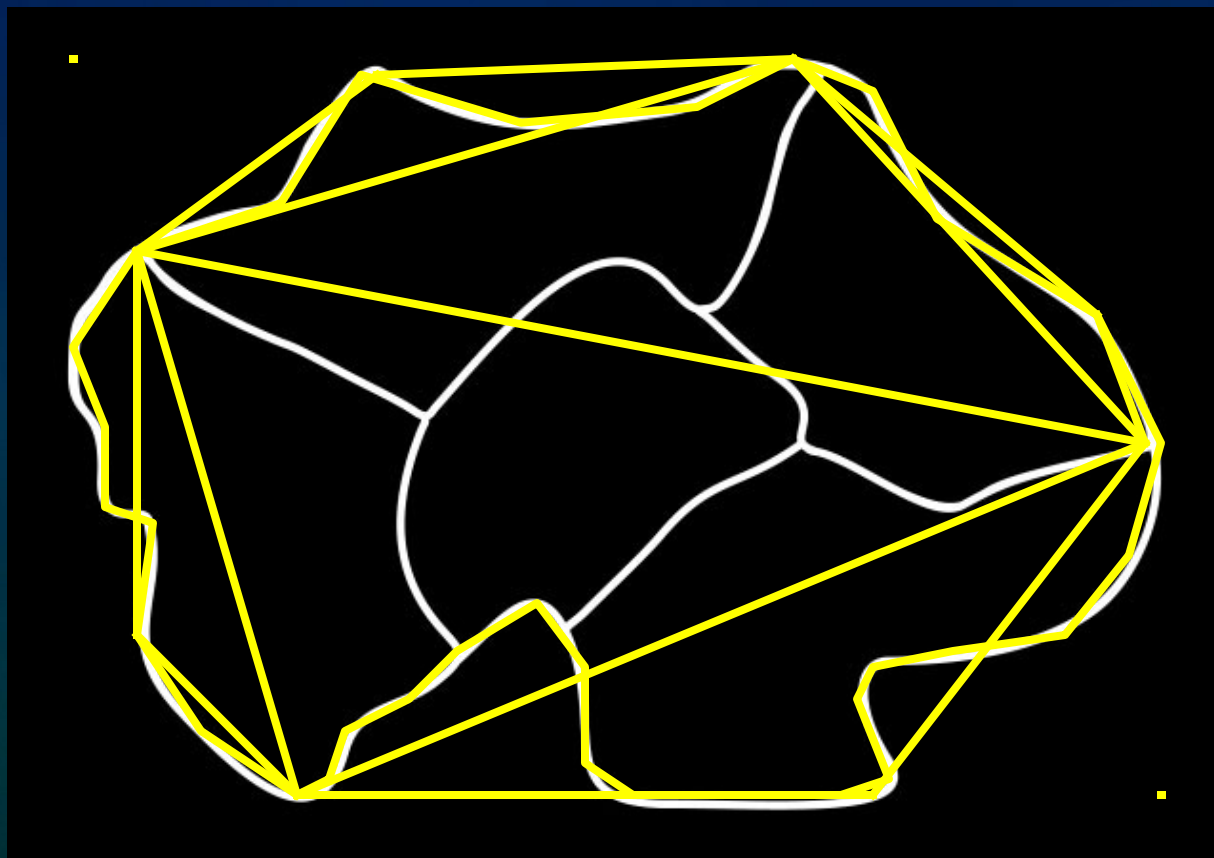
Object Representation

Fit Triangular Mesh to selected Object
- for efficient OpenGL Rendering



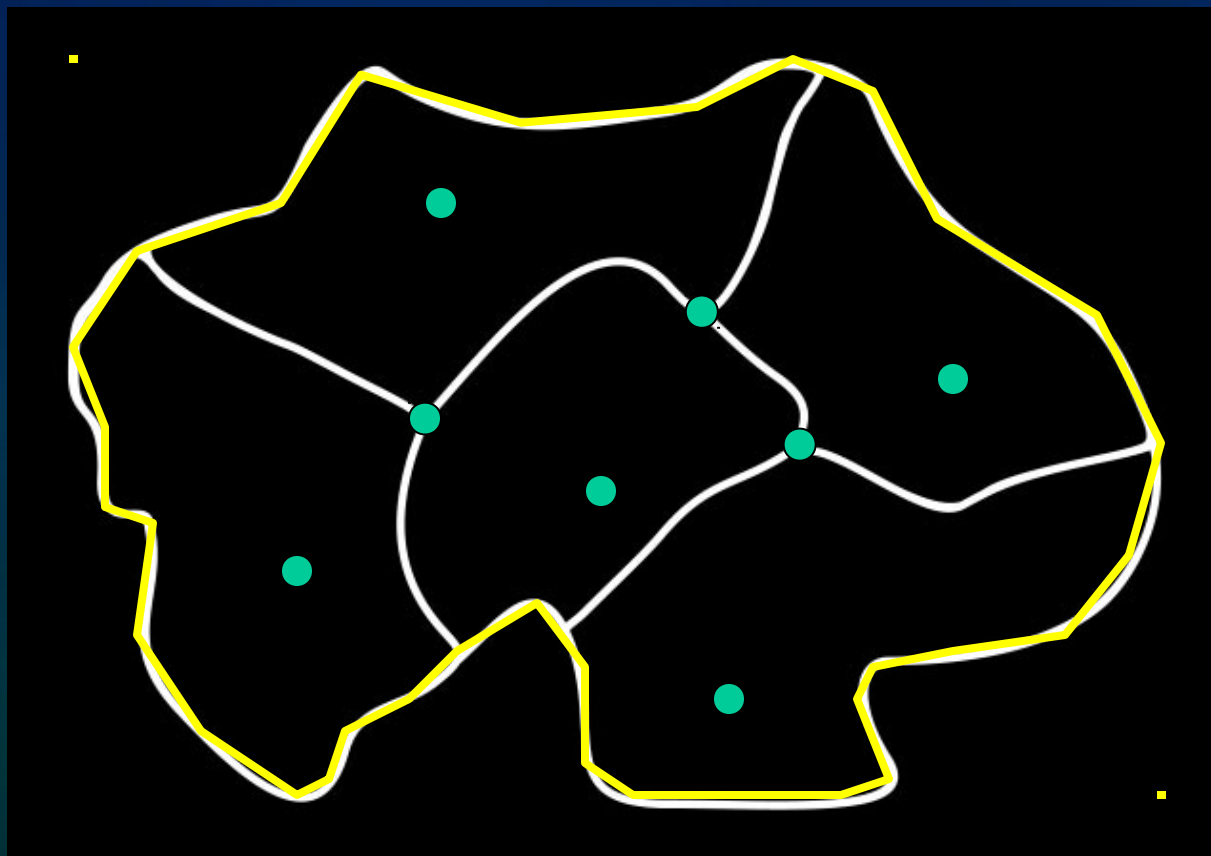
Object Representation

1. Recursive Divide & Conquer Polygonalization of Object Boundary



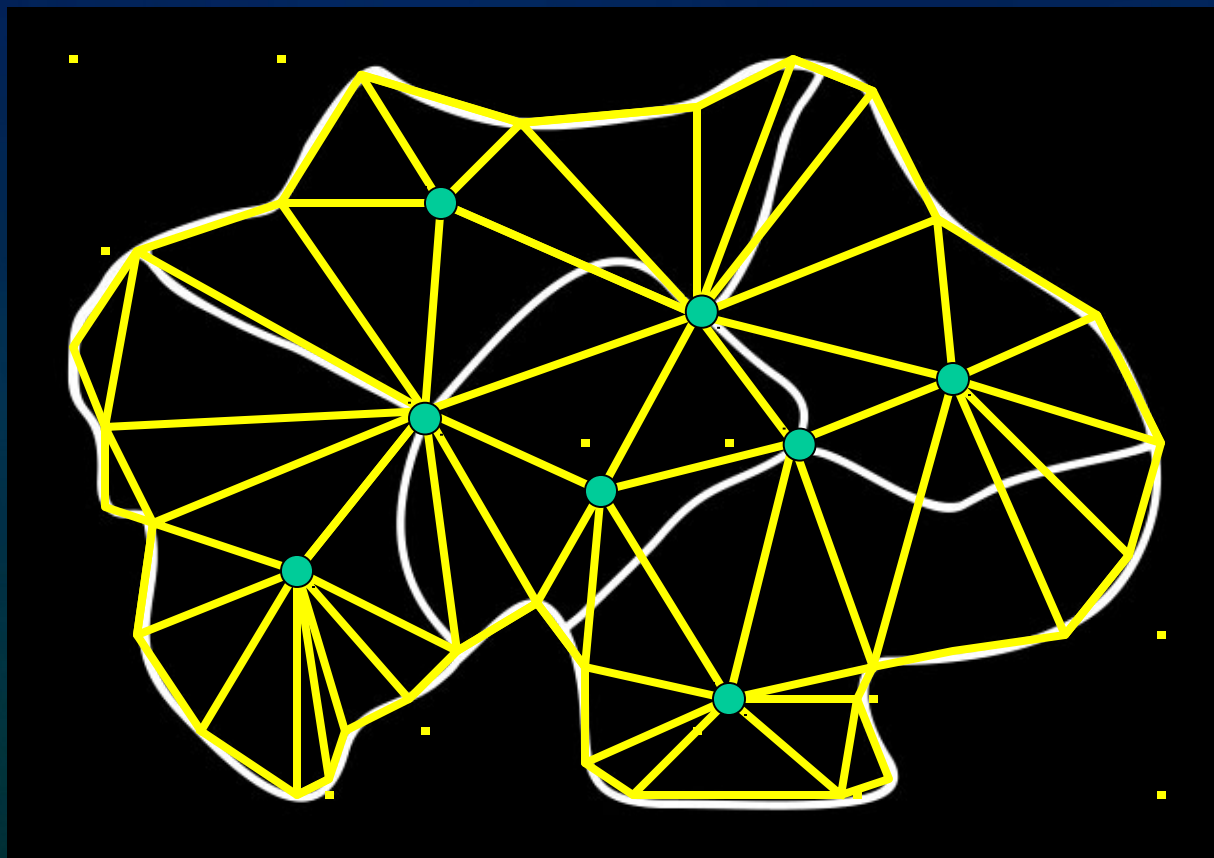
Object Representation

2. Insert Nodes (corners and basepoints)



3. Final Triangulation

- using Delaunay algorithm





Object Editing Operations

1. Object selection & representation
2. Object editing operations
3. Rendering
4. Background filling and texture painting
5. Applications



Object Editing Operations

2. Object editing operations
 - a. Direct Editing
 - b. Indirect Editing



Object Editing Operations

2. Object editing operations
 - a. Direct Editing
 - b. Indirect Editing



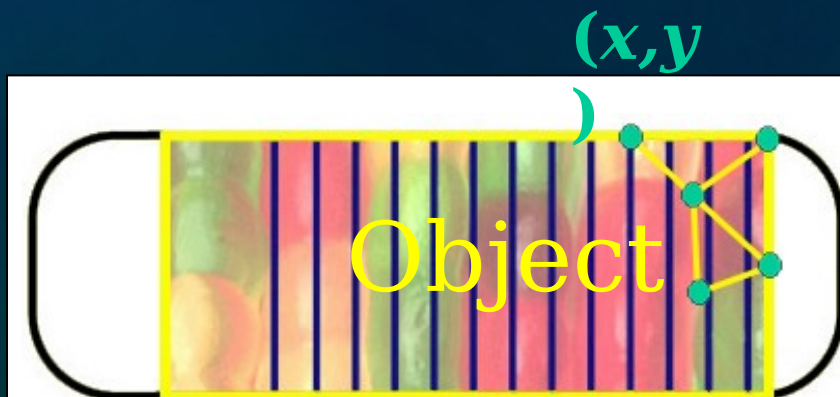
Stretch

1. After the **object** has been selected



Stretch

1. After the **object** has been selected
2. And **vertices** (x,y) identified





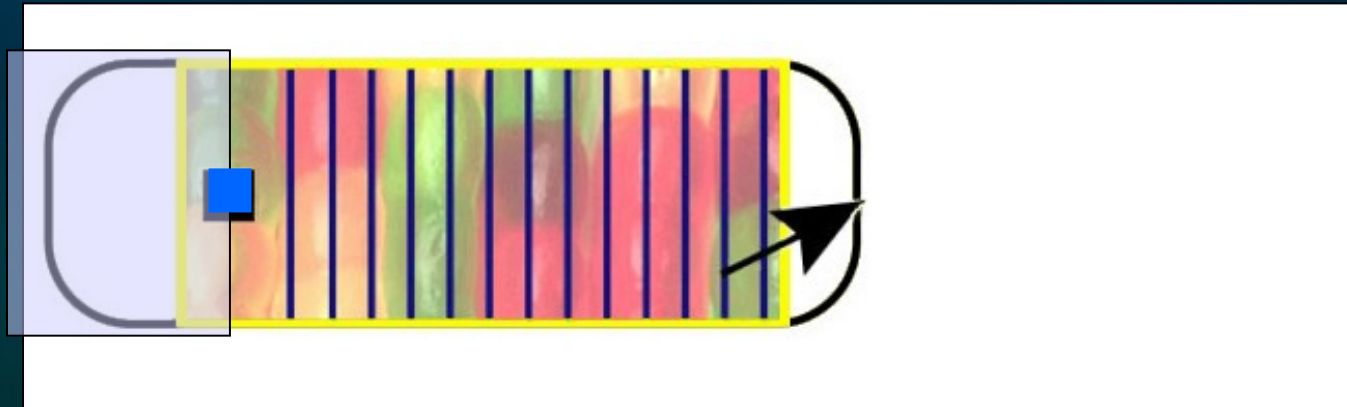
Stretch

1. After the **object** has been selected
2. And **vertices** (x,y) identified
3. An **anchor/pivot** point is specified



Stretch

1. After the **object** has been selected
2. And **vertices** (x,y) identified
3. An **anchor/pivot** point is specified
4. And the user clicks on or near the **object**
stretch it with respect to the **anchor**



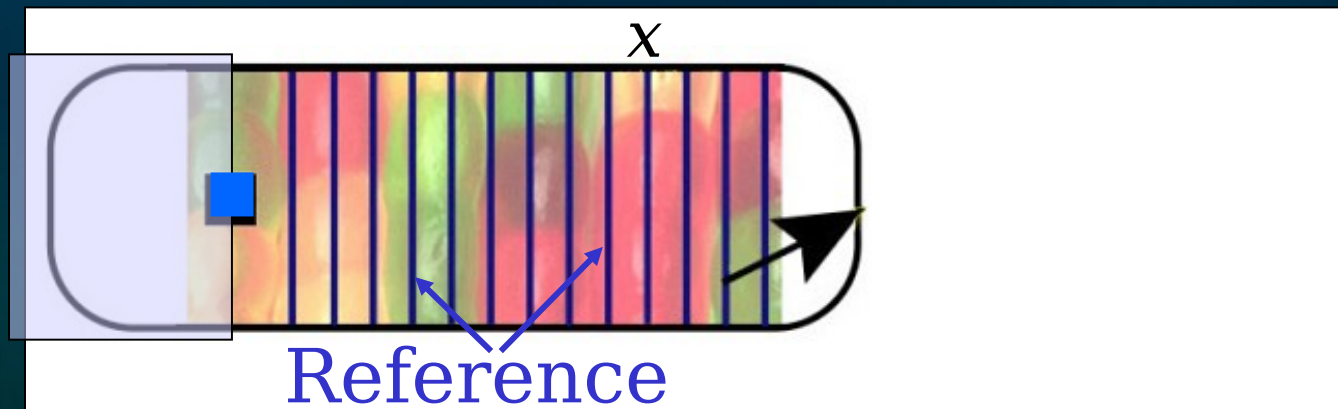


Stretch

Object vertices (x,y) , are stretched

$$x' = x \left(1 + \frac{\Delta x}{b_x} \right)$$

$$y' = y \left(1 + \frac{\Delta y}{b_y} \right)$$



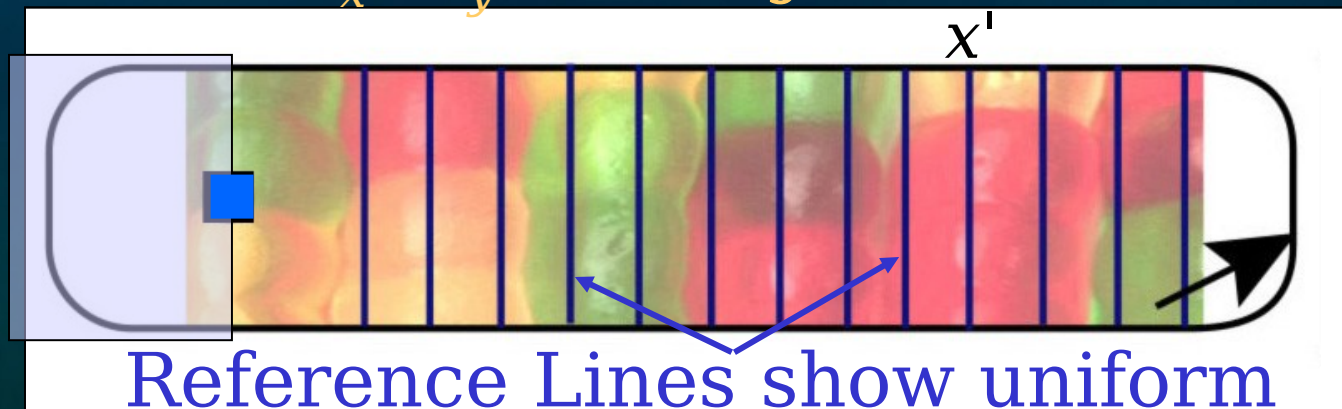
Stretch

Object vertices (x,y) , are stretched to (x',y') ,

$$x' = x \left(1 + \frac{\Delta x}{b_x} \right) \quad y' = y \left(1 + \frac{\Delta y}{b_y} \right)$$

where $(\Delta x, \Delta y)$ captures cursor movement

b_x, b_y are object dimensions

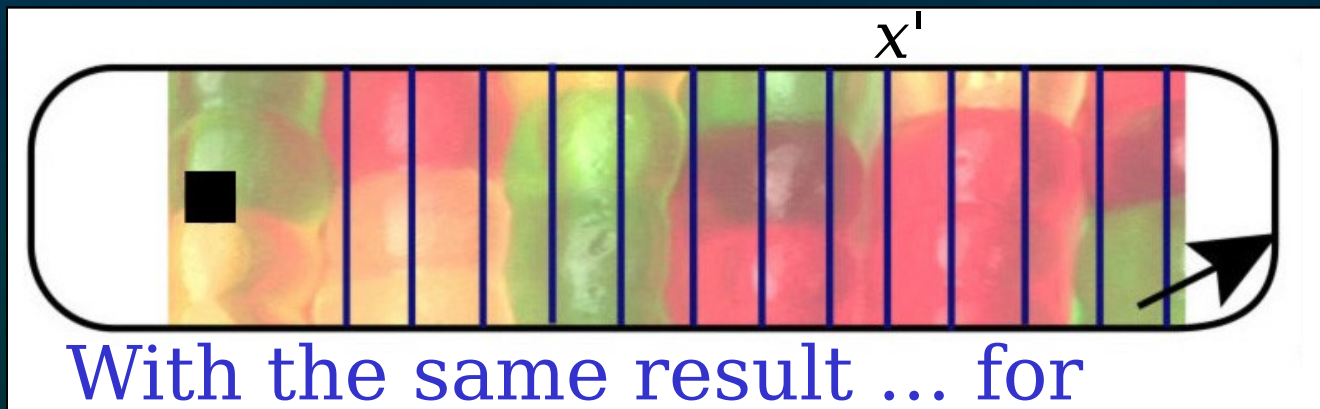
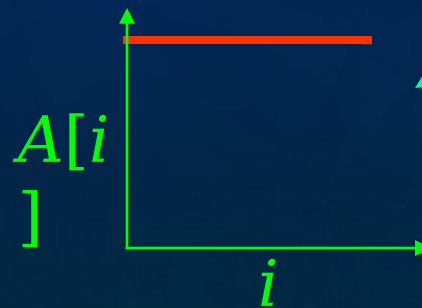




Simple Linear Stretch

We now introduce an attenuation function, $A[i]$

$$x' = x \left(1 + A[i] \frac{\Delta x}{b_x} \right) \qquad y' = y \left(1 + A[i] \frac{\Delta y}{b_y} \right)$$



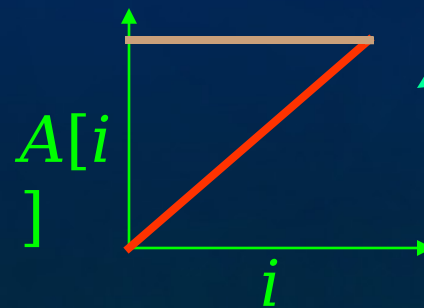


Nonlinear Stretch

SAN ANTONIO
SIGGRAPH
2002

But if we vary $A[i]$

$$x' = x \left(1 + A[i] \frac{\Delta x}{b_x} \right) \quad y' = y \left(1 + A[i] \frac{\Delta y}{b_y} \right)$$





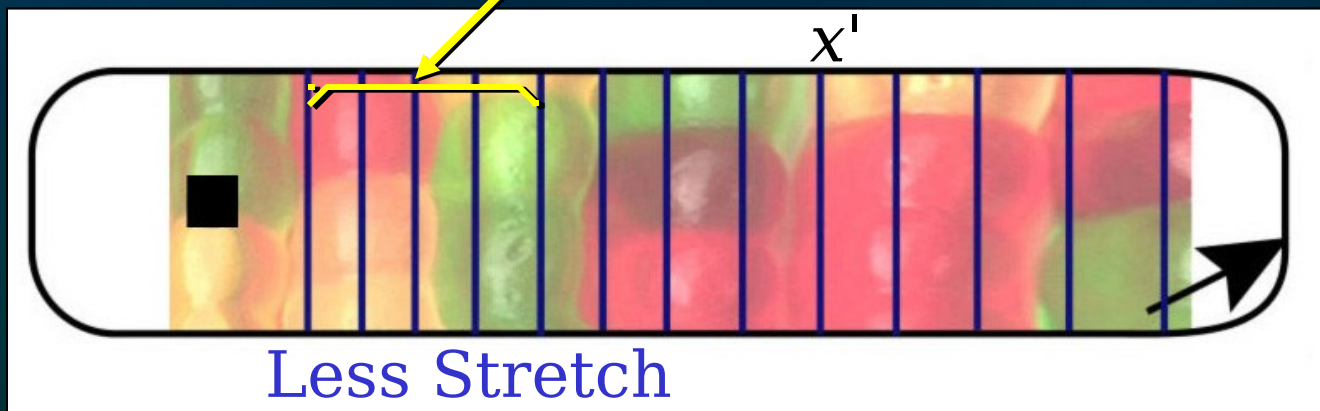
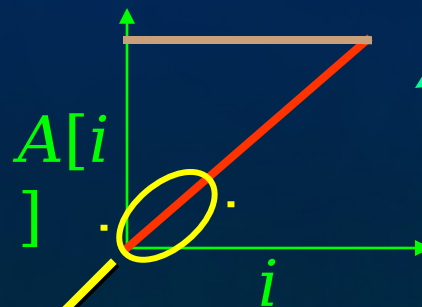
Nonlinear Stretch

SAN ANTONIO
SIGGRAPH
2002

But if we vary $A[i]$

$$x' = x \left(1 + A[i] \frac{\Delta x}{b_x} \right)$$

$$y' = y \left(1 + A[i] \frac{\Delta y}{b_y} \right)$$





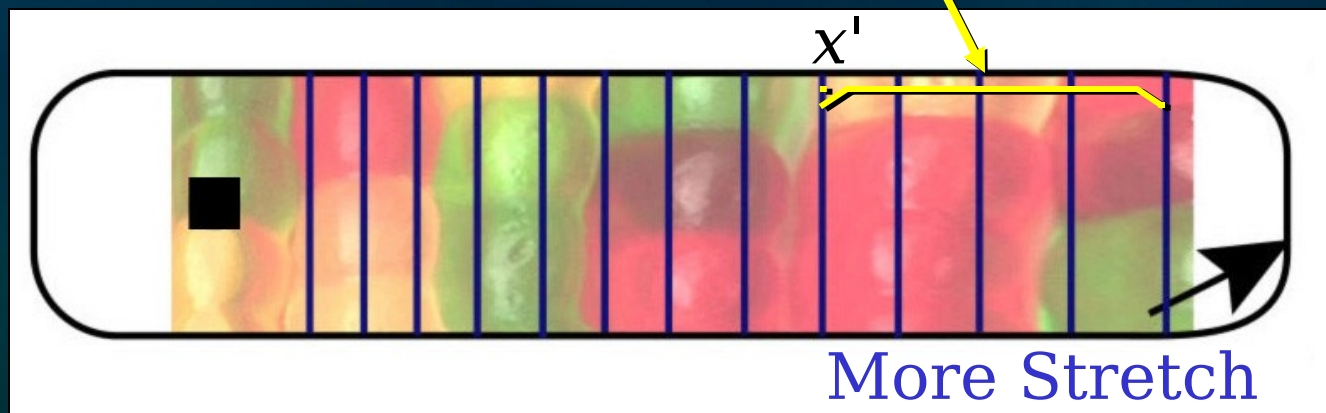
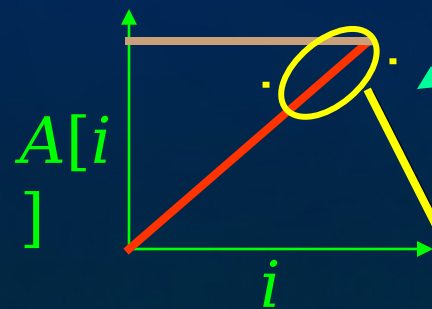
Nonlinear Stretch

SAN ANTONIO
SIGGRAPH
2002

But if we vary $A[i]$

$$x' = x \left(1 + A[i] \frac{\Delta x}{b_x} \right)$$

$$y' = y \left(1 + A[i] \frac{\Delta y}{b_y} \right)$$



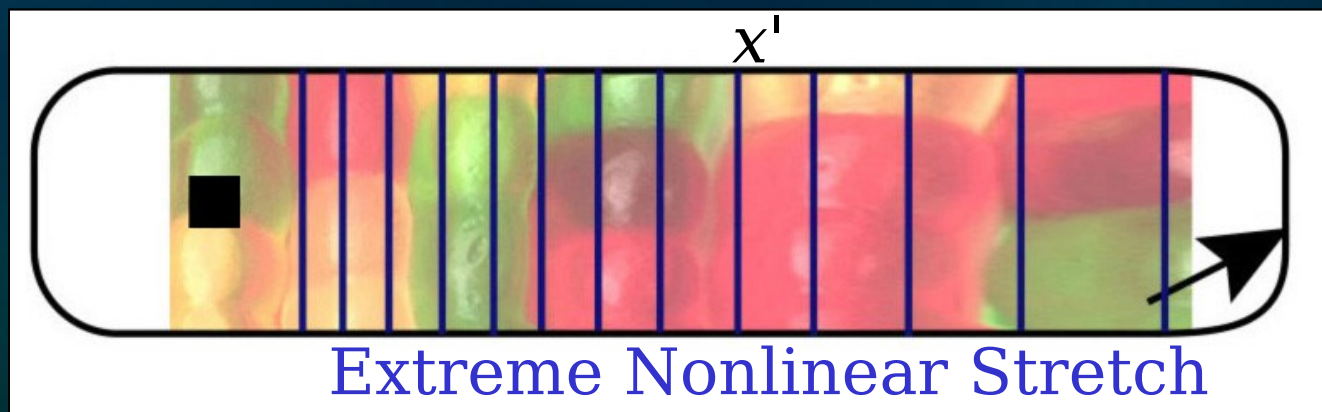
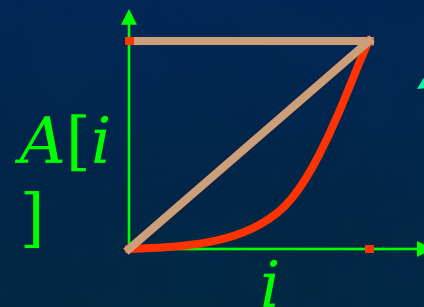


Nonlinear Stretch

SAN ANTONIO
SIGGRAPH
2002

And if we vary $A[i]$ even more

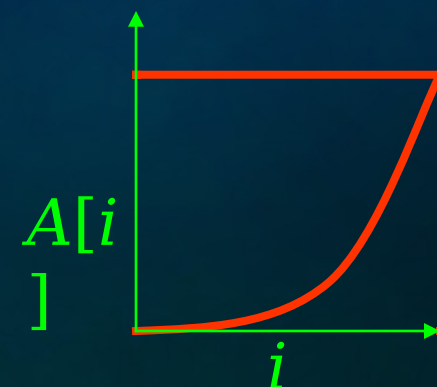
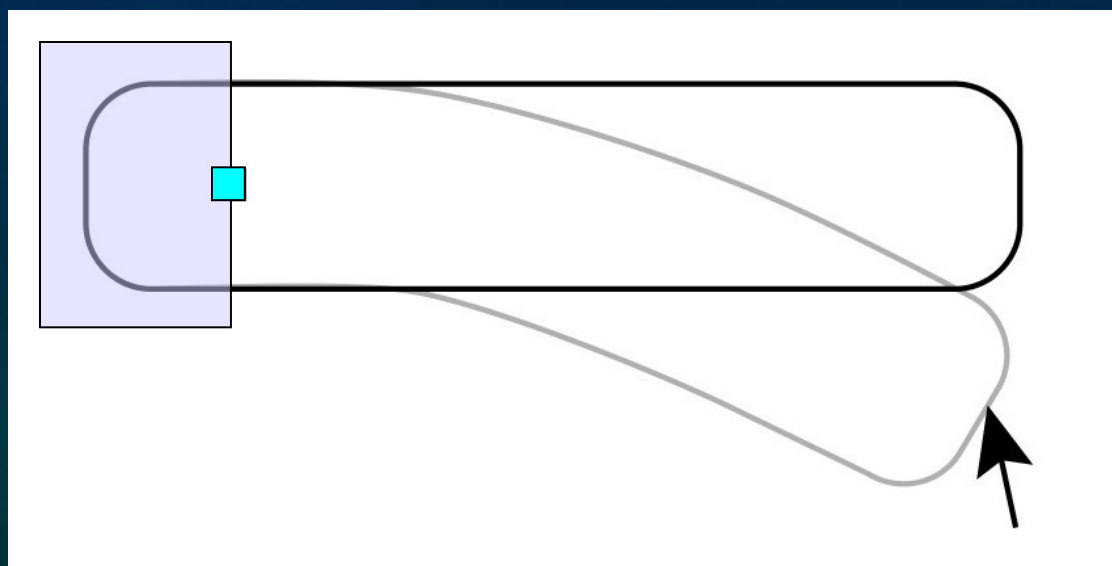
$$x' = x \left(1 + A[i] \frac{\Delta x}{b_x} \right) \quad y' = y \left(1 + A[i] \frac{\Delta y}{b_y} \right)$$



Rotate

Simple rotation matrix, with attenuated θ

$$(x', y') = \begin{bmatrix} \cos(\theta A[i]) & -\sin(\theta A[i]) \\ \sin(\theta A[i]) & \cos(\theta A[i]) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$





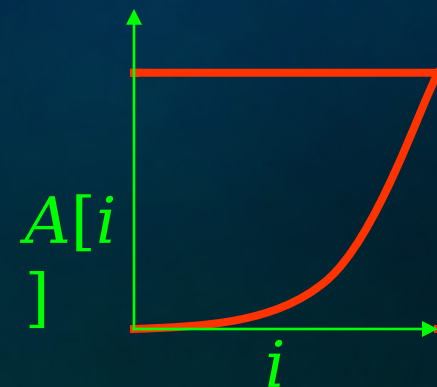
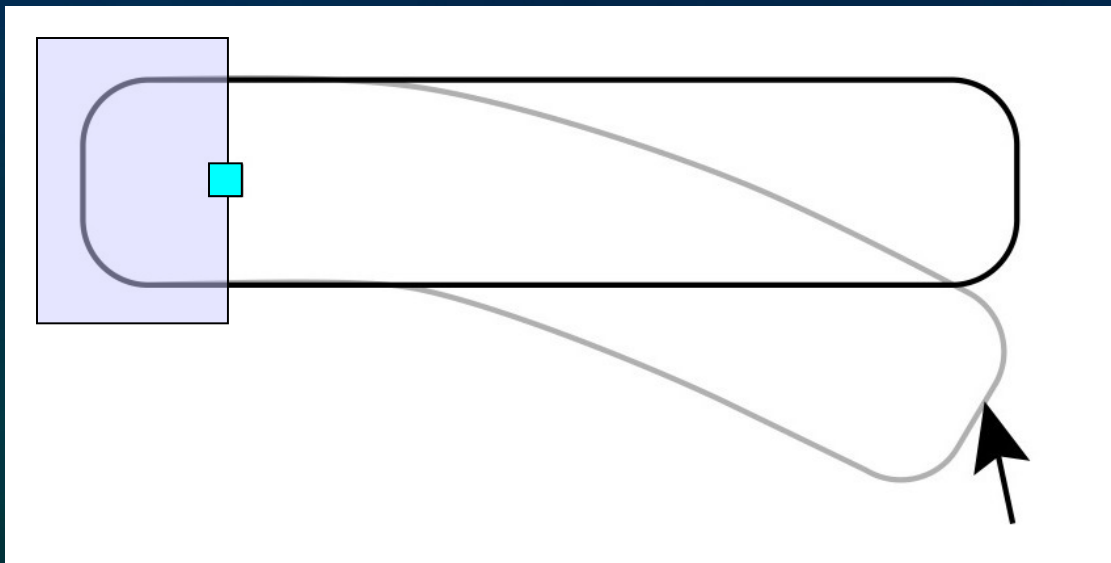
Rotate
Bend

Rotational

SAN ANTONIO
SIGGRAPH
2002

Simple rotation matrix, with attenuated θ

$$(x', y') = \begin{bmatrix} \cos(\theta A[i]) & -\sin(\theta A[i]) \\ \sin(\theta A[i]) & \cos(\theta A[i]) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

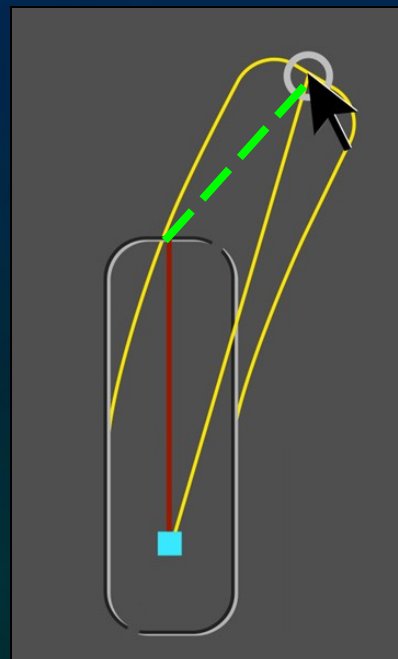


Bend-Stretching

- Calculate the bend based on stretch

$$x' = x \left(1 + A_l[i] \frac{v_n - v_o}{b_x} \right) \quad y' = y$$

$$(x'', y'') = \begin{bmatrix} \cos(\theta A_r[i]) & -\sin(\theta A_r[i]) \\ \sin(\theta A_r[i]) & \cos(\theta A_r[i]) \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$



User
Actions
Computati
on



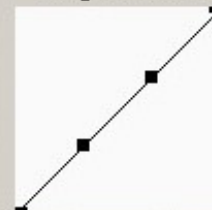
Indirect Editing

2. Object editing operations
 - a. Direct Editing
 - b. Indirect Editing

Curve Deformers - Defaults

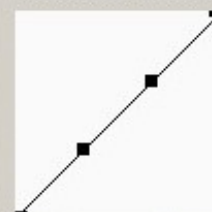


Length Stretch



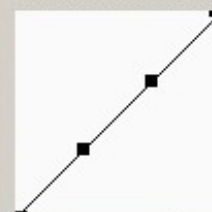
Reset

Thickness Stretch



Reset

Rotation Falloff



Reset

OK

Length

Thickness

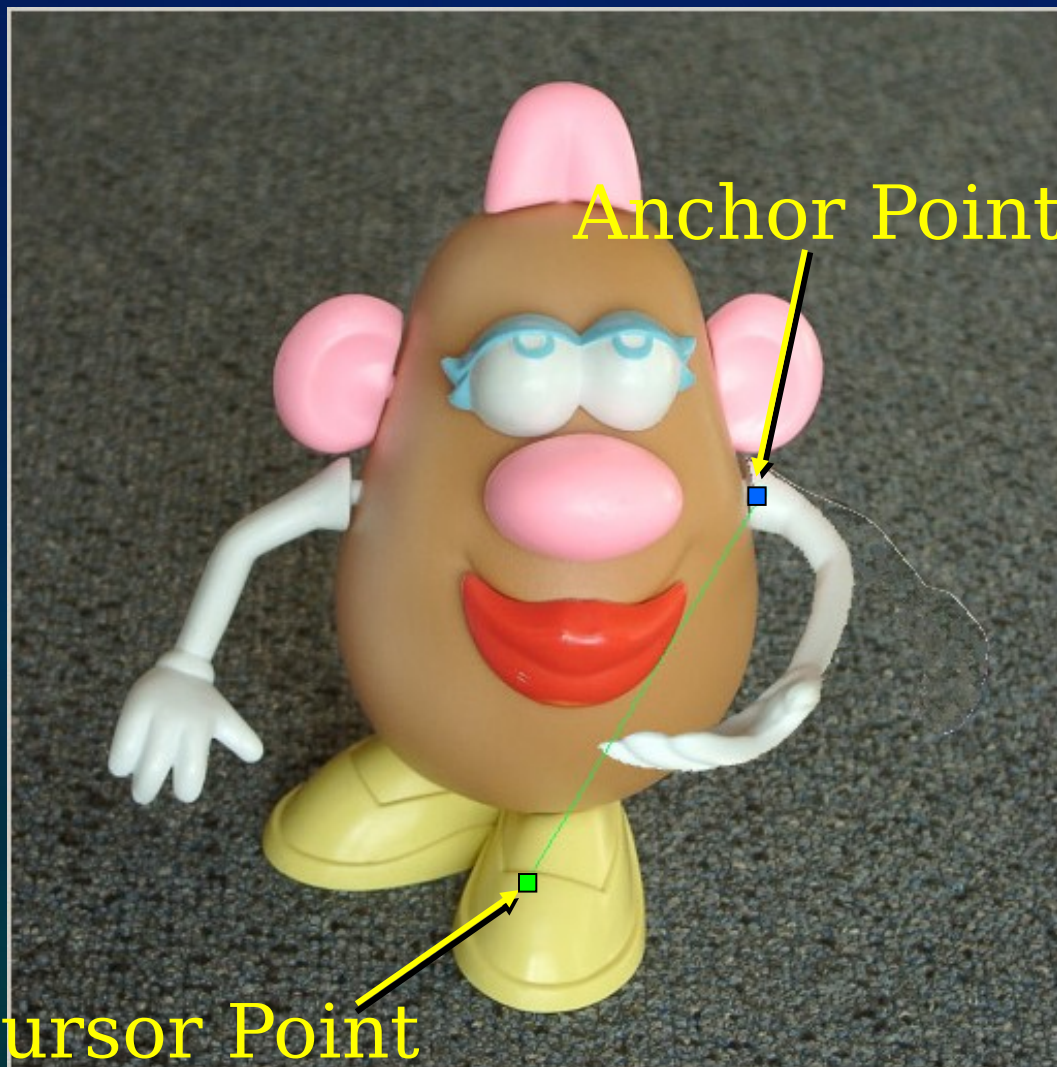
Rotation

Cursor Point



Length Flattening

SAN ANTONIO
SIGGRAPH
2002

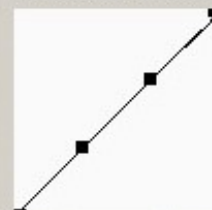


Length Stretch



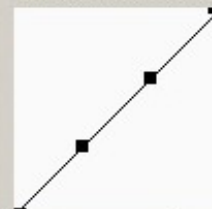
Reset

Thickness Stretch



Reset

Rotation Falloff



Reset

OK

**Length
~ flat**

Thickness

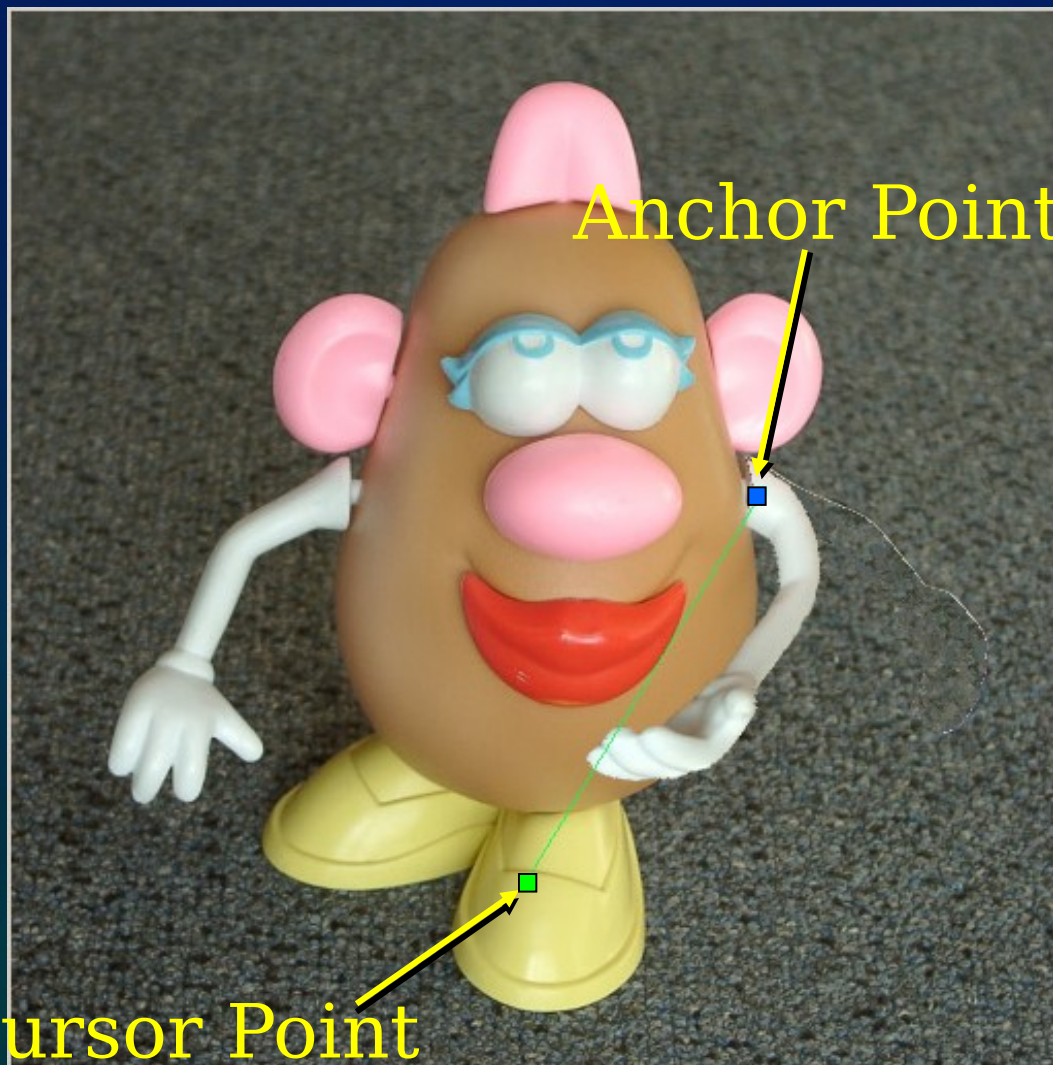
Rotation

Cursor Point



Bend-Stretching

SAN ANTONIO
SIGGRAPH
2002

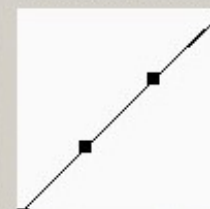


Length Stretch



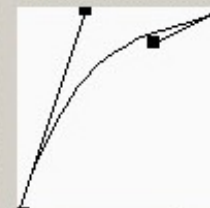
Reset

Thickness Stretch



Reset

Rotation Falloff



Reset

OK

**Length
Leveling**

Thickness

**Rotation
Level-of**



Indirect Editing - Interactive

SAN ANTONIO
SIGGRAPH
2002

Object Based



Rendering

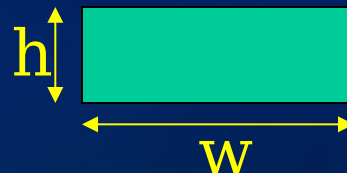
1. Object selection & representation
2. Object editing operations
3. **Rendering**
4. Background filling and texture painting
5. Applications



Antialiasing using OpenGL

Variables:

- Layer Height
- Layer Width

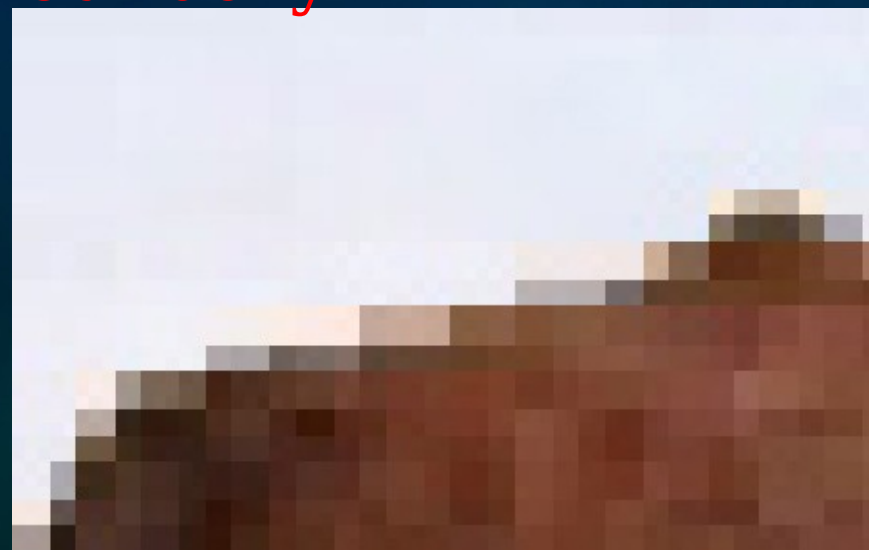
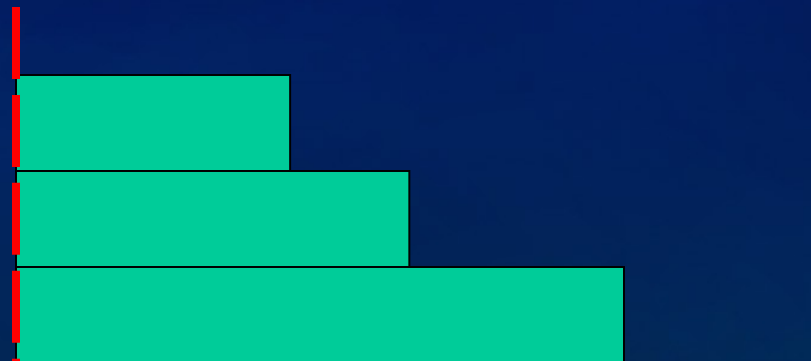


Antialiasing using OpenGL

Variables:

- Layer Height
- Layer Width
- Number of Layers

Object
Object
Boundary





Background Filling

1. Object selection & representation
2. Object editing operations
3. Rendering
4. Background filling and texture painting
5. Applications

Background Filling

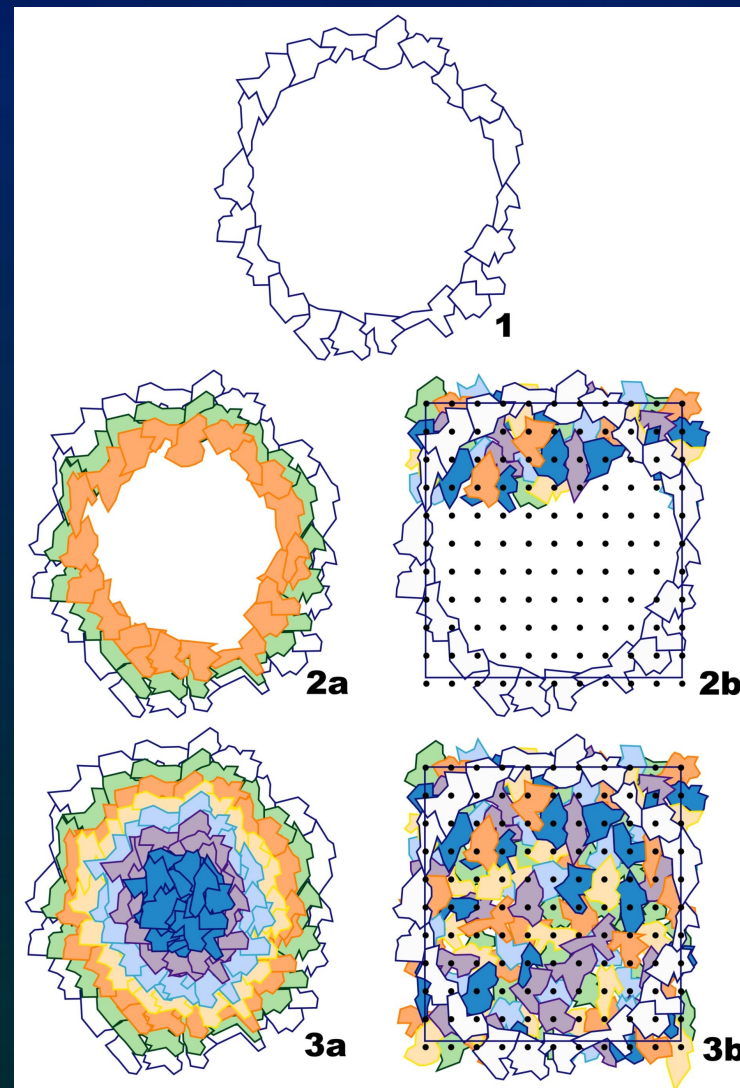
2 Methods

1. Scale Down
Filling (2a-
3a)

2. Gridded (2b-
3b)

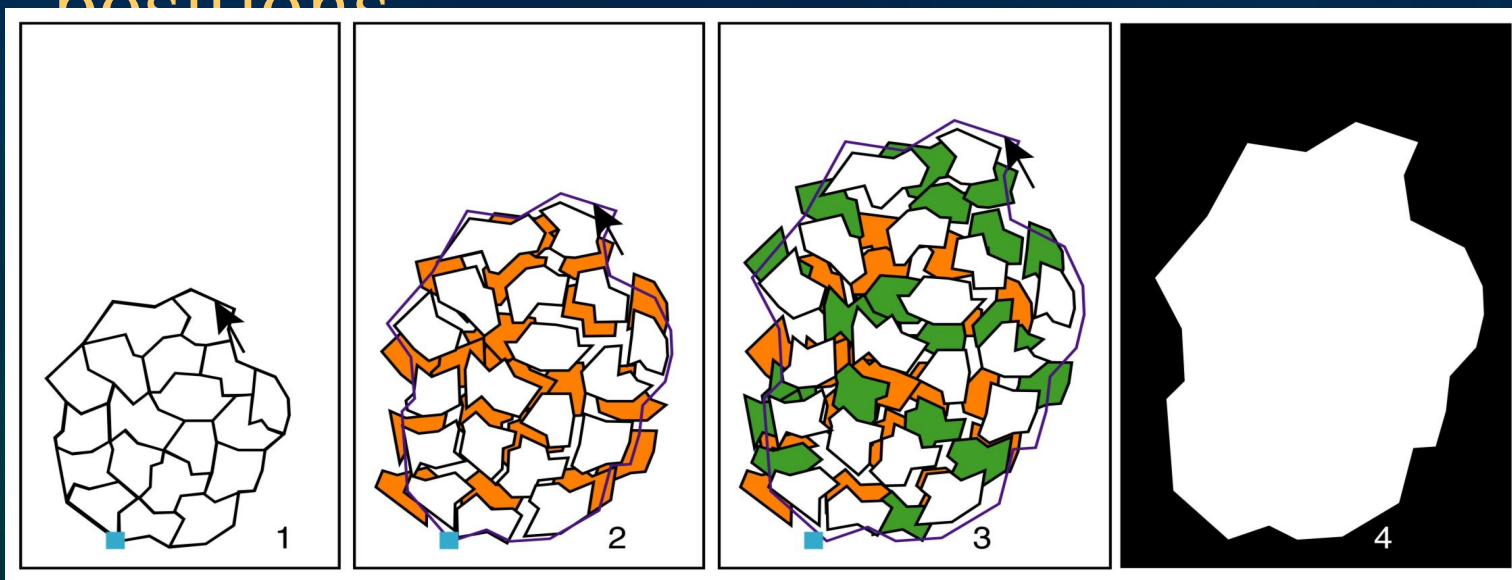
Sampling

1. Automatic



Texture Preservation

- When an object stretches, if its texture is high in detail, the texture becomes overly smoothed, looking unnatural.
- To fix this, we keep object TRAP sizes constant, warping only their basepoint positions.





Applications

1. Object selection & representation
2. Object editing operations
3. Rendering
4. Background filling and texture painting
5. Applications



Video



Contributions & Conclusions

1. New object selection tool
2. Trap-based triangulation
3. Automatic background filling
4. Attenuation functions $A[i]$
5. Curve deformers
6. OpenGL antialiasing
7. Texture painting
8. Texture preservation
9. Image editing at the object level



Future Work

1. Structured background filling
2. Intelligent selection tools
(Intelligent Scissors, Paint)
3. Predefined/Advanced/Compound
Attenuation functions
4. Subpixel edge model for
decontamination of object
fringe
5. Use multiple anchor points,